Authors: *Yusuke Sugihara - Maria Camila Millan Toro*

Data Analytics
Course Assignment N.07: 20 New Channels
Academic Year 2023

INTRODUCTION

In this project, we will examine a set of documents with their word occurrence patterns. The dataset is divided by train and test and in each there are 20 folders that represent the 20 different sources. The goal is to understand and describe the data, convert it into word feature vectors, and finally cluster the documents based on the occurrence patterns of words and find the categories that described the data most. This means we will perform a text clustering task.

NOTE: Since most of the analysis is done with natural language processing, text analysis, and clustering techniques, we considered a better option for data visualization using Python packages *Seaborn, sklearn.datasets, Matplotlib, etc* rather than *Tableau*. Also, our dataset is exactly the same as the **newsgroups-dataset**, we will use the ['The sklearn.datasets.fetch_20newsgroups function'] so that we can efficiently analyze the dataset.

DATA ANALYSIS

a) DATA PREPROCESSING

Data parsed.
We retrieved the data from the online dataset that contains identical information to the one we previously downloaded. The dataset consists of over 18,000 newsgroup documents that are classified into 20 different groups, each representing a distinct topic. Some newsgroups are interconnected (such as c comp.os.ms-windows.misc/comp.windows.x), while others are vastly dissimilar (such as rec.autos/talk.religion.misc). The following is a collection of the 20 newsgroups, categorized (to some extent) by their subject matter:

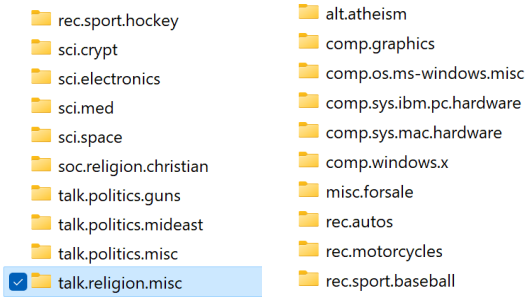


Figure 1. Categories of the dataset.

To understand better the dataset we are dealing with, we checked how many files are within the 20 folders. There are 18846 documents including the train (11314) and the test (7532) datasets. All these files have the structure of an email text file as shown in Figure 2.

```
From: iacs3650@Oswego.EDU (Kevin Mundstock)
Subject: Joe Robbie Stadium "NOT FOR BASEBALL"
Reply-To: iacs3650@oswego.Oswego.EDU (Kevin Mundstock)
Organization: Instructional Computing Center, SUNY at Oswego, Oswego, NY
Lines: 16

Did anyone notice the words "NOT FOR BASEBALL" printed on the picture
of Joe Robbie Stadium in the Opening Day season preview section in USA
Today? Any reason given for this?

Also, I just noticed something looking at the Nolan Ryan timeline in
the preview. On 8/22/89, Rickey Henderson became Nolan's 5000th strikeout.
On 6/11/90 he pitched his 6th no-hitter against Oakland. I believe the
last out in the game was made by Rickey Henderson. And on 5/1/91, Nolan
pitched his 7th no-hitter on the same day a certain someone stole his
939th base, which overshadowed it. It seems that Nolan is having a lot of
publicity at Rickey's expense. IMO, Rickey deserves it, and it seems as
most of the net agrees with me from what I've seen on it lately. They are
both great players, but IMO, Nolan has outclassed Rickey, both in playing
and more importantly, in attitude. Just my thoughts.

                                    Kevin
```

Figure 2. Example of one of the emails.

Data cleaning
In this phase, the purpose is to make the dataset persistent to make a text representation and to eliminate words that are not relevant for clustering purposes. Hence, text preprocessing was done by performing the following steps:

1. Removing headers, footers, and quotes. We used the function remove from the library fetch_20newsgroup, so 'headers' removes information about the sender's name or email address, 'footers' removes blocks at the ends of posts that look like signatures, and 'quotes' removes lines that appear to be quoting another post.

2. Replacing words: As in this type of text (emails) there are a lot of contractions, we decided to construct a small dictionary to replace those and have them as the general form. For example: "won't", "will not", "cannot", "can not", "can't", "can not","n't", " not", what's", "what is", "it's", "it is", "%", " percent".

3. Removing stop words (words like articles, prepositions, pronouns, conjunctions, etc. which do not add much information about the topic of the text like "is", "which"," the", "and" "of, etc.). For this, we used the function "STOPWORDS" from the library "wordcloud" and we also added more words that should be counted as stop words like: 'from, 'also', 'even', 'thing', 'first', 'much', 'many', 'last', 'just', 'etc', 'anything', 'something', 'my', 'mine'.

4. Removing punctuation and special characters ("*", "*", "!","?") by using the function string.punctuation that includes many built-in punctuation characters.

5. Removing digits. For instance: >16µsec will be replaced with "micro sec"

6. Tokenization is for breaking down the string or the text of each file into smaller units called tokens (which are individual words). For this purpose, we used the function tokenize of the NLKT library. For example, the text "Note that this effect is not caused by anything actually on the SCSI Bus" was tokenized as "Note", "that", "this", "effect", "is", "not", "caused", "by", "anything", "actually", "on", "the", "SCSI", "Bus".

7. Lemmatization for noun, adjective, verbs and adverbs, used to reduce those into the base or dictionary form or known as a lemma. For example, the words "eats", "eaten", "eating" and "ate" will be transformed into the general form of the verb "eat". We applied the WordnetLemmatizer which is a class provided by the NLTK (Natural Language Tool Kit) library in Python.

Below, there is an example of the function implemented to preprocess the data with one of the documents (emails) of the training dataset.

Raw Text:

From: gnelson@pion.rutgers.edu (Gregory Nelson)
Subject: Thanks Apple: Free Ethernet on my C610!
Article-I.D.: pion.Apr.6.12.05.34.1993.11732
Organization: Rutgers Univ., New Brunswick, N.J.
Lines: 26
Well, I just got my Centris 610 yesterday. It took just over two weeks from placing the order. The dealer (Rutgers computer store) appologized because Apple made a substitution on my order. I ordered the one without ethernet, but they substituted one _with_ ethernet. He wanted to know if that would be "alright with me"!!! They must be backlogged on Centri w/out ethernet so they're just shipping them with! Anyway, I'm very happy with the 610 with a few exceptions. Being nosy, I decided to open it up _before_ powering it on for the first time. The SCSI cable to the hard drive was only partially connected
(must have come loose in shipping). No big deal, but I would have been pissed if I tried to boot it and it wouldn't come up! The hard drive also has an annoying high pitched whine. I've heard apple will exchange it if you complain, so I might try to get it swapped. I am also dissappionted by the lack of soft power-on/off. This wasn't mentioned in any of the literature I saw. Also, the location of the reset/interupt buttons is awful. Having keyboard control for these functions was much more convenient. h, and the screen seems tojump in a wierd way on power-up. I've seen this mentioned by others, so it must be a...feature... Anyway, above all, it's fast. A great machine at a great price!
gnelson@physics.rutgers.edu

After performing data cleaning:

well get centris yesterday take two weeks place order dealer rutgers computer store appologized apple make substitution order order one without ethernet substitute one with ethernet want know would alright me must backlogged centri wout ethernet ship with anyway happy exceptions nosy decide open before power time scsi cable hard drive partially connect must come loose ship big deal would piss try boot would come up hard drive annoy high pitch whine hear apple exchange complain might try get swap dissappionted lack soft poweronoff mention literature saw also location resetinterupt button awful keyboard control function convenient oh screen seem tojump wierd way powerup see mention others must a feature anyway all fast great machine great price.

NOTE: We decided to use only Lemmatization instead of Stemming because the first one aims to transform the words into their base or dictionary form and it considers the linguistic context instead of doing generalizations, so the performance of this method is more accurate, preserving the semantic meaning of the words, better feature representation, which is extremely important due to the clustering task.

b) DATA VECTORIZATION

Data vectorization, for converting raw data into a numerical representation suitable called "vector", which is an array or list of numerical values. The process of data vectorization involves transforming the original data into a structured numerical format while preserving the relevant information. This process helps in reducing the dimensionality of the data, improving computational efficiency, and making the data suitable to apply machine learning algorithms as well as data analytic tasks. In this project, we used the function "TfidfTransformer" of the library sklearn.feature_extraction.text to convert the raw processed document into a matrix of TF-IDF features. This helps to weight words higher that only occur frequently in a specific document while generally frequent words are weighted lower or even canceled out if they occur in every document. After ignoring terms that appear in more than 70% of the documents (as set by max_df=0.7) and terms that are not present in at least 10 documents (set by min_df=5), the resulting number of unique terms n_features is around 16,000. We can additionally quantify the sparsity of the X_tfidf matrix as the fraction of non-zero entries divided by the total number of elements. We find that around 0.6% of the entries of the X_tfidf matrix are non-zero.

c) DATA EXPLORATORY

-How many documents are in each topic?

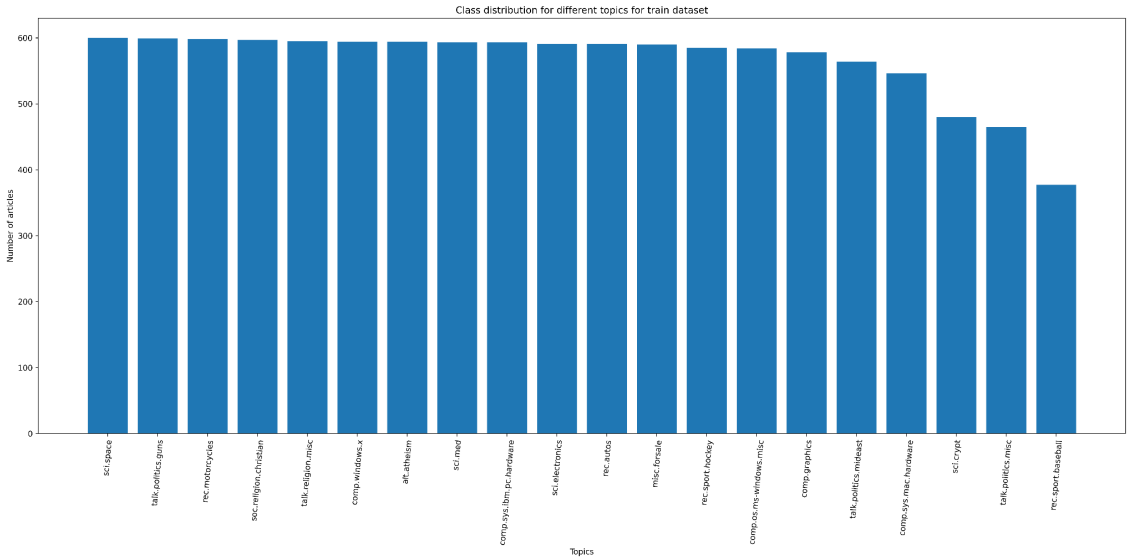


Figure 3. Class distribution of the training dataset.

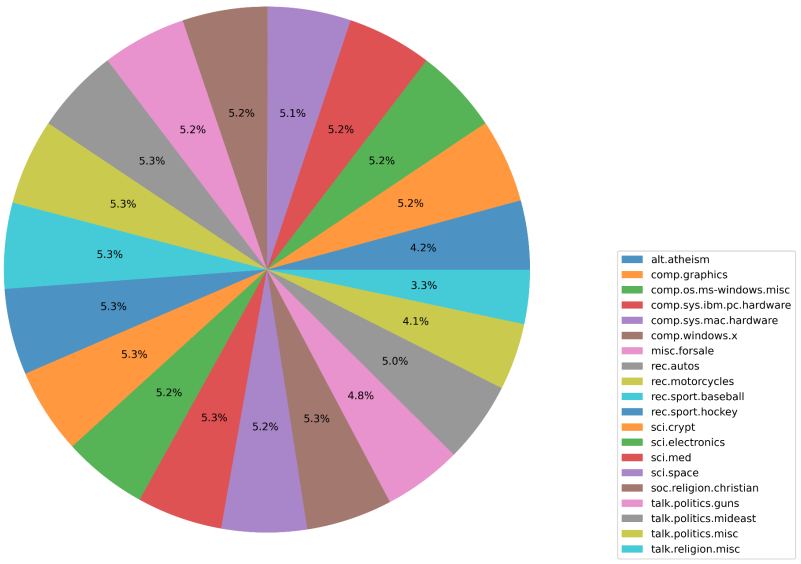


Figure 4. Class distribution for the different topics of the dataset.

As we can see in the previous graphics, the documents for each category are evenly distributed among 20 topics which means that there is not a high imbalance within the data. There are 11314 documents within the 20 folders or categories. We could have also ignored the last 3 categories (sci.crypt, talk,politics.misc, rec.sport.baseball) as the distribution is slightly different from the others, but we decided to keep them as the distribution difference was not high enough to make changes in the data behavior.

-What are the most common words along the whole training dataset?

To verify the effectiveness of the preprocessing, we utilized the wordcloud package to generate a visual depiction of the frequently occurring words. This step was crucial in comprehending the data and confirming the progress made thus far. Additionally, it helped identify whether further preprocessing or data cleaning was required before proceeding with model training. The following figure shows that the top 10 of the most common words along the whole train dataset are: dollar, percent, year, well, people, want, good, take, right, and file.

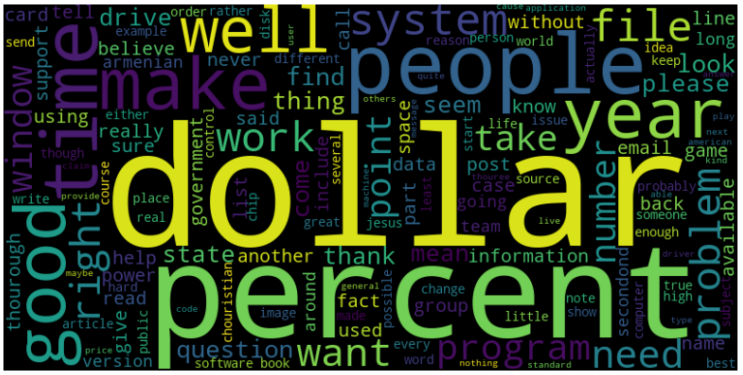


Figure 5. Class distribution for the different topics of the dataset.

-Frequency distribution of the training dataset

In order to understand, identify patterns of the data and prove that the pre-processing data was done correctly, we checked the frequency distribution of words for the whole training dataset as well for 3 categories. Figure 11 represents an overview of the 30 most common words within the 11314 documents, some of those words are dollar, percent, one, get, use, and people.

- For the category called “alt.atheism”, the frequency distribution has 9599 samples and 1714162 outcomes, where the most common words are “god” with almost 1600 times, “believe”, “Jesus”, “atheism”, “atheist”, “argument” and “bible with around 200 word counts..
- For the category called “comp.graphics”, the frequency distribution has 10701 samples and 192581 outcomes, where the most common words are “image”, “file”, “program”, “software”, “system”, “package” and “email”.
- For the category called “comp.os.ms-windows.misc ”, the frequency distribution has 9599 samples and 1714162 outcomes, where the most common words are “dollar”, “window”, “percent”, “use”, “get”, “one” and “card”.

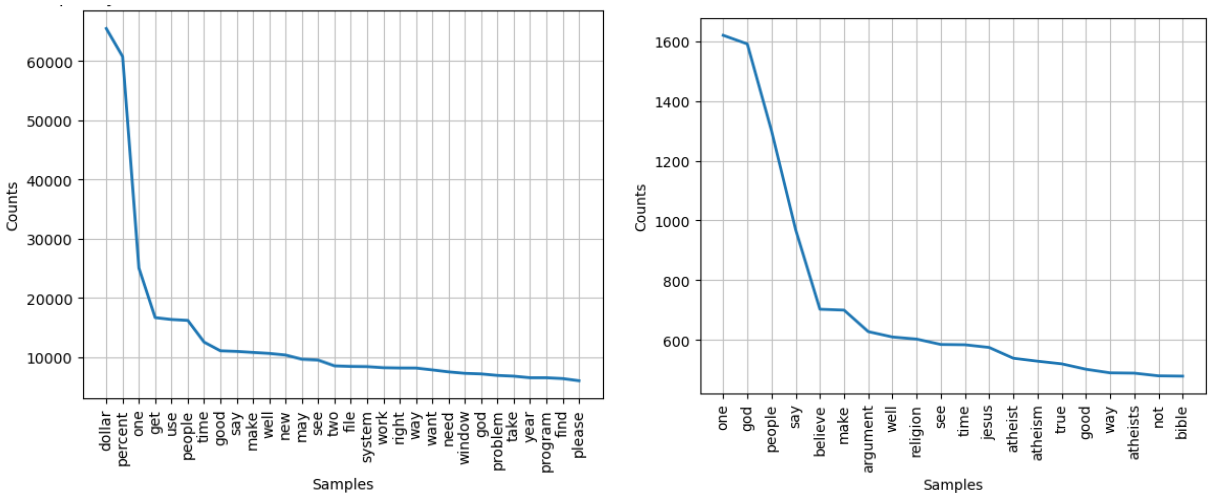
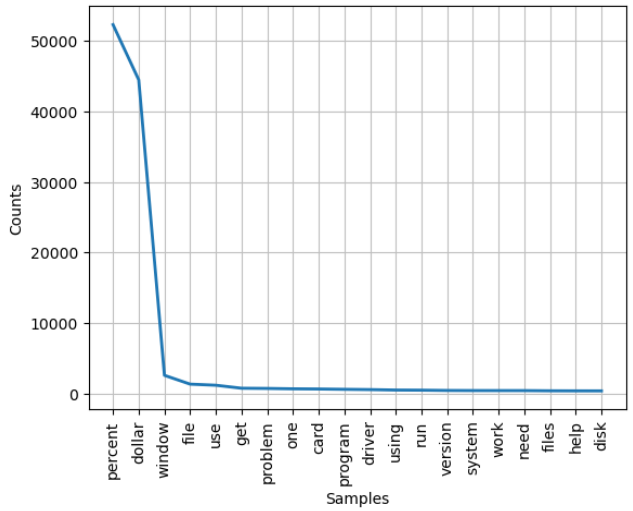


Figure 6. Frequency distribution for the different topics of the dataset. /Figure 7. Frequency distribution for the Category: alt.atheism.



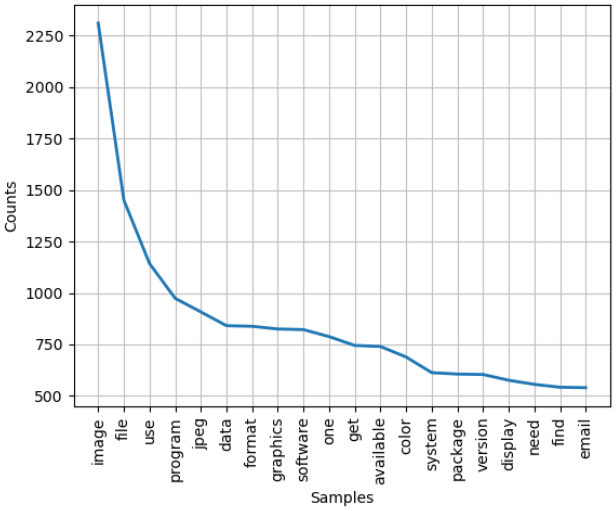


Figure 8. Frequency distribution for the Category: comp.graphics /Figure 9. Frequency distribution for the category: comp.os.ms-windows.misc

-Document lengths distribution

The distribution of document lengths is important for the text analysis, as we could understand the structure and the complexity of the text corpus, allowing us to identify how short, long or how is the variation of lengths for each document in the training dataset. Therefore, we loaded the dataset, calculated the word count for each document, and created a histogram plot to visualize this distribution.

-The average amount of words or the length that a document has is 374.
-The median or the second quarter is 160, which reflects the central tendency of the words.
- Most of the documents in the dataset have less than 500 words.
-The standard deviation is 1129, which is a considerably higher value that points that the dataset is more spread out from the mean because some documents have a vast amount of words having more than 2000 words.

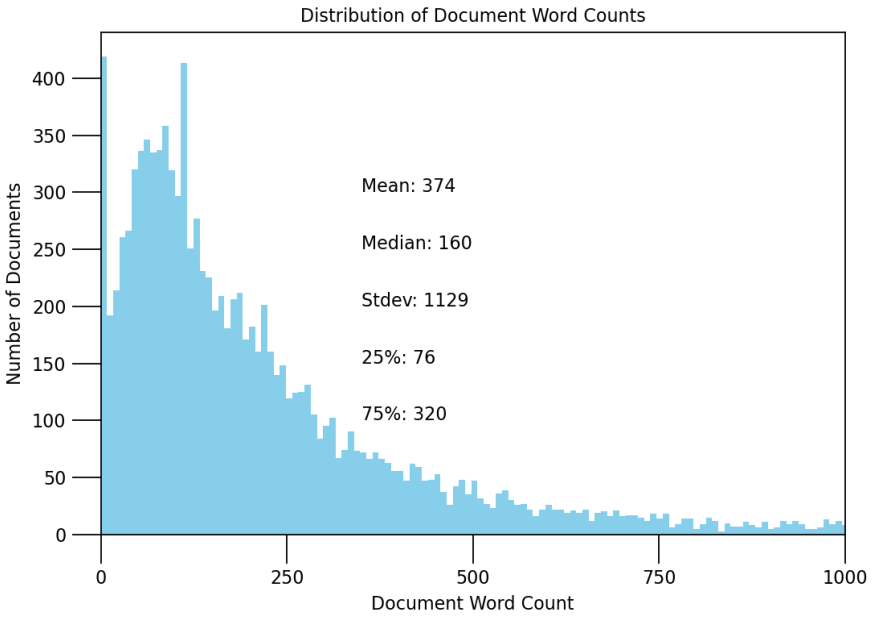


Figure 10. Distribution of document word counts of the training preprocessed dataset.

d) DATA CLUSTERING

Data clustering is a technique used in unsupervised machine learning to group similar data points together based on their inherent characteristics or patterns. The goal of data clustering is to discover clusters or groups of documents, which share words with similar topics or information related. The clusters are created based on the similarity of the data points in terms of their features or attributes. This is an unsupervised technique, but we have the labels to corroborate the efficiency of the clustering task. We implemented two clustering algorithms:

1. K-means

It aims to partition the data into a pre-defined number (k) of clusters, where each data point is assigned to the nearest cluster centroid. The algorithm minimizes the within-cluster sum of squared distances. In order to find the optimal number of clusters, we decided to use two techniques:

2. Minitbatch Kmeans

The Mini-batch K-means clustering algorithm is a version of the standard K-means algorithm in machine learning. It uses small, random, fixed-size batches of data to store in memory, and then with each iteration, a random sample of the data is collected and used to update the clusters. We used the function MiniBatchKmeans of the library sklearn.cluster, for which we have to specify the number of clusters to form as well as the number of centroids to generate.

Determining the optimal value of K, the number of clusters.

Now in order to identify which is the most suitable number of clusters for the K-means and Minibatch K-means methods, we decided to implement the following methods:

- **Elbow method for Kmeans and MiniBatch Kmeans**

The objective is to find the appropriate value of K that strikes a balance between minimizing the within-cluster sum of squares (WCSS) and avoiding overfitting. The elbow point signifies the point at which the improvement in cluster quality diminishes, indicating a trade-off between reducing WCSS and avoiding unnecessary complexity associated with increasing the number of clusters. With an increasing value of K, the WCSS tends to decrease as more clusters can better accommodate the data points. For K-means clustering, the optimal number of clusters might be 16, while the Minibatch K-means clustering algorithm performs best with 6 clusters. However, since the graph did not provide enough information to determine the appropriate number of clusters, an alternative method was employed to ascertain it.

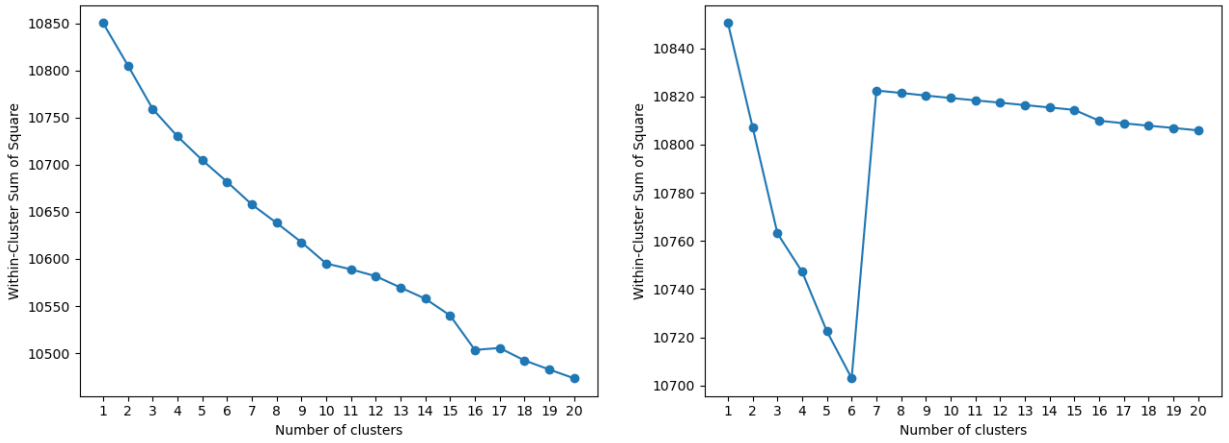


Figure 11. Elbow method representation for selecting the K cluster numbers for the K-means and the MiniBatch K-means clustering methods.

- **Silhouette score for Kmeans and Mini BatchKmeans**

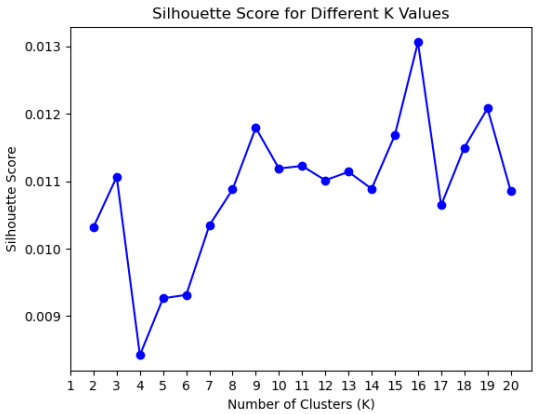


Figure 12. Silhouette score for selecting the K cluster numbers.

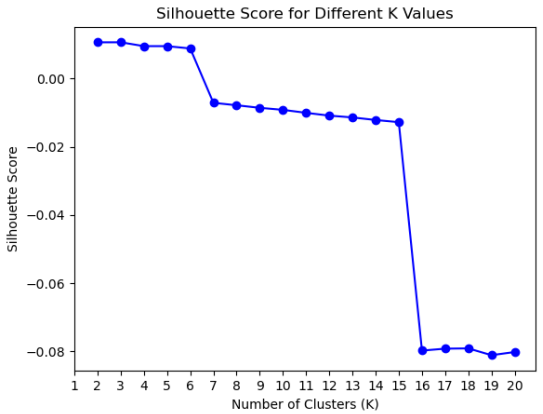


Figure 13. Silhouette score for selecting the K cluster numbers.

Silhouette analysis calculates the average silhouette coefficient for each K, which measures how close each sample in one cluster is to the samples in the neighboring clusters. From the following figure, we could infer that for this dataset, the most optimal number of clusters is 16 because it registers the highest score of the silhouette, which means better-defined and well-separated clusters. Similarly, for Minibatch Kmeans, the optimal number of clusters is 6 because the graph experienced a drastic decrease in terms of the Silhouette score of more than 6 clusters.

Based on the labels available in the training dataset, we could deduce that categories such as "talk.politcs.guns", "talk.politics.mideast" and "talk.politics.misc" could have been merged within one cluster, This behavior sounds logical because at reviewing the data, some categories are extremely similar to each other, which allows us to group them into one category and find the optimal k, number of clusters, which in this case is 16.

e) DIMENSIONALITY REDUCTION

KMeans and MiniBatchKMeans can cause the phenomenon called the Curse of Dimensionality for high dimensional datasets such as text data. Since the data we are working with is too large to calculate, we decided to perform a dimensionality reduction of the data by using LSA, which stands for Latent Semantic Analysis. For this purpose, we used the function TruncateSVD (Singular Value Decomposition), which works perfectly for the kind of data we are dealing with, TF-IDF matrices. When using LSA, the clustering time can be not only decreased but also the stability of the computation can be improved. This dimensionality reduction was done for the training dataset as well as for the test dataset. We wanted to check how much different the behavior of the two clustering methods with or without this dimensionality reduction (LSA).

f) EVALUATION OF THE MODELS ON THE TRAINING SET

Clustering algorithms are inherently unsupervised learning techniques. Nonetheless, in the case of having class labels for a particular dataset, it becomes possible to employ evaluation metrics that utilize this "supervised" ground truth information to measure the effectiveness of the generated clusters. The metrics that we considered for evaluating both clustering techniques (K-means and MiniBatch K-means) are the following:

- Homogeneity, which quantifies how much clusters contain only members of a single class
- Completeness, which quantifies how many members of a given class are assigned to the same clusters
- V-measure, the harmonic mean of completeness and homogeneity
- Rand-Index, which measures how frequently pairs of data points are grouped consistently according to the result of the clustering algorithm and the ground truth class assignment
- Adjusted Rand-Index, a chance-adjusted Rand-Index such that random cluster assignment have an ARI of 0.0 in expectation.

Therefore, we created a function to fit and evaluate these metrics to both models and to the training and test datasets as well.

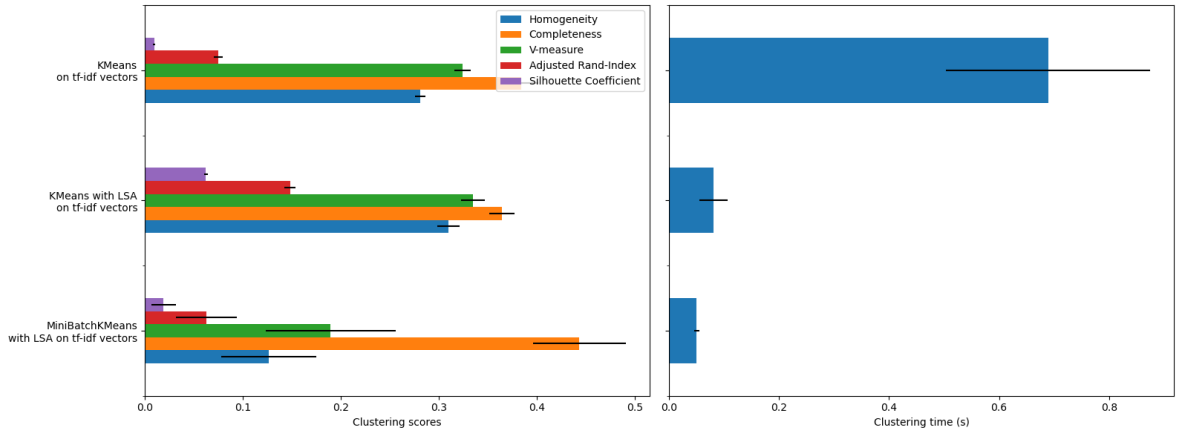


Figure 14. Comparison of the two clustering methods selected: K-means and MiniBatch Kmeans.

From the previous figure, we could infer the following:

- Minibatch K-means with LSA is the fastest among the other techniques used in this project. This is consistent with our intuition because Minbatch Kmeans utilizes small random batches of data of a fixed size so those data can be stored in memory and also LSA reduces the dimension of the dataset to calculate.

- Even though minibatch K-means can provide faster convergence and scalability benefits, it sacrifices some clustering accuracy compared to the traditional K-means.
- It can be said that there is a trade-off relationship between the accuracy of clustering and the speed of convergence.
- K-means with LSA on tf-idf vectors outperform other models. It has better results overall compared to the other models, with good accuracy and with less computational time.

In summary, considering the importance placed on accuracy, we have determined that K-means with LSA is the best model.

g) DATA VISUALIZATION

Top Terms per cluster (K means with LSA)

Since TfidfVectorizer can be inverted, we can identify the cluster centers, which provide an intuition of the most influential words for each cluster. Below, the list of the most frequent words per cluster is presented with the selected number of clusters 16:

Cluster 1: file program use server image code color problem widget files
Cluster 2: dollar sale offer price shipping new sell condition email best
Cluster 3: drive card scsi video disk mhz mac bus problem controller
Cluster 4: know article post say good read really want make sure
Cluster 5: thanks email advance know thank information info address help reply
Cluster 6: time space use know good way problem power really little
Cluster 7: window file use program manager problem application version know run
Cluster 8: bike ride motorcycle dod bikes honda helmet riding road miles
Cluster 9: game team year play season win league hockey players player
Cluster 10: people say evidence true believe church christian claim point bible
Cluster 11: car cars engine speed good dealer drive power price know
Cluster 12: people israel gun government state armenian israeli law make time
Cluster 13: new york brand know post old want make group people
Cluster 14: right people rights amendment government left law militia state wrong
Cluster 15: god jesus bible christ faith believe christian people say sin
Cluster 16: key chip encryption clipper government phone use escrow algorithm keys

From the previous output, we can infer that the clustering process was done correctly as the words for each cluster are correlated with one specific topic. For instance. for cluster 10, we could say the documents are talking about religion, which matches with the category that we have in the dataset, called “talk.religion.misc”. Now, for cluster 15, we have words like “Jesus”, “Christ”, and “Christian” which means they could be related to the category “soc.religion.christian”.

Word Cloud
Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Therefore, a word cloud is generated for each cluster, visually representing the words within the cluster with varying font sizes. The font size of a word in the word cloud indicates its importance or frequency within that specific cluster. Words that are more significant or commonly occurring in the cluster are displayed with larger font sizes, while less important or infrequent words are represented with smaller font sizes. For instance, when examining cluster 8, key terms like “bike”,“motorcycle”, “honda” and “helmet” emerge as significant words. This indicates that this particular cluster can be associated with vehicles. Similarly, focusing on cluster 15, prominent words in the word cloud include “god”, “christ” and “belief”. This suggests that this cluster is related to religion.

Figure 15. The 20-word clouds with K-means clustering (k=16).

h) EVALUATION OF THE SELECTED MODEL (k-MEANS) WITH THE TEST DATASET.

We chose the K-means with the LSA algorithm as it was the one that showed the best accuracy and we evaluated it in the test dataset.

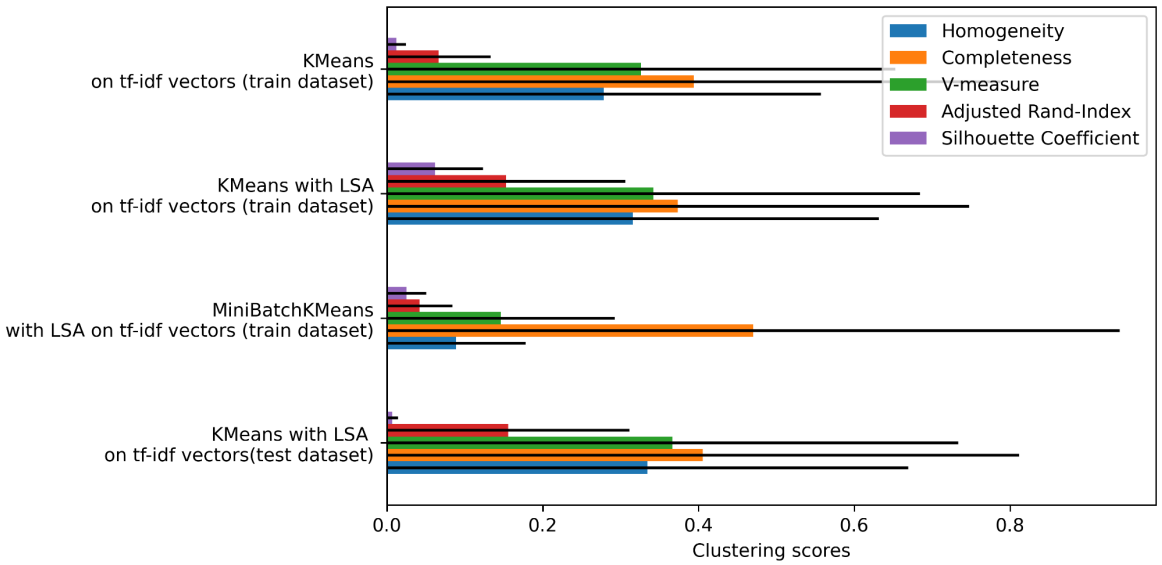


Figure 16. The 20-word clouds with K-means clustering (k=16).

Top Terms per cluster (K means with LSA)

We applied the clustering method selected, K-means with LSA (k=16), to the training dataset in order to check the performance of the clustering model. Below, there is the list of the most common 10 words per cluster:

Cluster 1: god jesus sin christian christ church bible edu christianity people
Cluster 2: indiana ucs edu silver creps irvine university writes article psych

Cluster 3: game team games edu hockey baseball espn year fans don
Cluster 4: se ericsson __ ___ 10 sweden edu fi 1993 apr
Cluster 5: nasa gov space edu jpl shuttle jsc sky world don
Cluster 6: au australia edu oz monash windows university cc deakin uwa
Cluster 7: sale car drive edu scsi new 00 com ide distribution
Cluster 8: muslims armenian muslim armenians serbs turkish war bosnian genocide serdar
Cluster 9: fbi edu koresh batf israel people com government children waco
Cluster 10: windows file edu dos com program thanks help use problem
Cluster 11: edu people don like just know com think good time
Cluster 12: com netcom writes article hp sun edu ibm posting nntp
Cluster 13: access digex net henry pat toronto communications hst express prb
Cluster 14: uk ac university demon writes article nz computer edu com
Cluster 15: edu university posting host nntp cs article writes state know
Cluster 16: ca canada bnr university carleton com writes bc baden freenet

From the above result, we can find that some of clusters created by the test datasets are clearly separated each other such as Cluster 5, which could be related to “sci.space”, and Cluster 8, which could be considered as “talk.religion.misc”. But also we could say that there are clusters similar to each other, specifically Cluster 14 and Cluster 15. They seem to be considerably similar to each other because both clusters have common words such as “edu”, “article” and “university” as significant words. Despite the possibility of associating Cluster 9 with the category "talk.politics," it is challenging to pinpoint the specific sub-topic within this cluster due to the lack of significant words.

CONCLUSIONS

In this project, we were able to develop methods for cleaning the dataset and performing tokenization and vectorization (TF-IDF) in order to have the data suitable for applying clustering methods. We determined the optimal number of clusters with the Elbow method and Silhouette score. We conducted a comparative analysis of document clustering using the K-means and the MiniBatch K-means clustering model. We selected the best model based on the accuracy and the computational efficiency, which K-means with LSA, and check the results of the top terms and word cloud depending on each cluster. Now, from the data itself, we can say there is some overlap between the clusters of different topics. This is because there are common words that are present in the documents that belong to different topics. For instance, when we consider the categories “talk.religion.misc” and “soc.religion.christian”, they are highly likely to have shared words with one another such as god, jesus, bible, christ, faith, and sin.

Bibliography

1. Choosing the right estimator: https://scikit-learn.org/stable/tutorial/machine_learning_map/
2. Datasets handling: https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html
3. Data clustering https://www.oneai.com/learn/text-clustering
4. Stemming and lemmatization: https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html
5. Minibatch K-means:https://www.geeksforgeeks.org/ml-mini-batch-k-means-clustering-algorithm/

Reference

1. https://github.com/rshah204/Text-Analytics-20-Newsgroups-Dataset/blob/master/1.%20ProjectCode.ipynb
2. https://github.com/hardikasnani/classifying-and-clustering-the-newsgroups/blob/main/classify-cluster.ipynb
3. https://github.com/Sameeksharajsb/20-Newsgroup-Dataset-Analysis/blob/main/Final%20Submission/Codes/FinalProject_codes1.ipynb